Cross-Layer Design: A Case for Standardization

Jeffrey Mehlman
Electrical Engineering Department, Stanford University
Stanford, California 94305
jmehlman at stanford dot edu

Index Terms—**Cross-Layer Design, Cross-Layer Framework, Cross-layer optimization, Wireless Sensor Networks, Distributed Source Coding, Energy efficiency, network lifetime, Data gathering, Communication Protocol, Open System Interconnection (OSI) Model**

I. INTRODUCTION

Recent work in wireless sensor networks (WSNs) has led to increasing utilization across a broad range of markets. Sensor networks, generally speaking, are systems comprised of a multitude of spatially-distributed nodes. Each node is generally capable of sensing some environmental parameter and communicating the data back to a central location. Some developing applications include military battlefield intelligence, biomedical patient monitoring, and smart buildings. A simple example of a ubiquitous sensor network is the security system found in many buildings today.

The functionality of a WSN node is generally implemented through electronics for sensing, intelligence (microcontroller), communication (radio) and a limited power source (battery). The communication system, utilized to transmit information between the nodes, is a major functional block in every WSN. Practical communication system design is aided by a well-defined conceptual framework called the Open System Interconnection (OSI) reference model. The OSI model is a seven-layer architecture, where each layer is responsible for specific sub-systems. The inter-layer interactions are strict, minimal, and well-defined.

Recent research has shown that the OSI model is not necessarily the correct approach for some modalities of wireless systems. Researchers have made modifications to communication protocols which violate the OSI model, but achieve specific optimization goals. These modifications are termed "cross-layer design (CLD)" [1]. A number of general approaches to CLD, often called "Cross-Layer Frameworks," (CLF) have been proposed, with none gaining widespread acceptance. Wireless communications design is diverging from the OSI model, but there is still no standard framework for CLD [1].

As discussed in [2], the lack of a standard framework can lead to many problems. This leads to reduced overall network performance. There are also fundamental, unanswered questions for CLD. Generally speaking, it is not clear when, where, and how different CLD proposals should be implemented.

This paper is organized as follows. Section II describes the background of communication protocol design, beginning with the OSI model. It identifies shortcomings of the OSI model for wireless communication systems, defines the need cross-layer design, and provides an example of how cross-layer design can improve wireless system performance.

Section III defines a methodology for evaluation of cross-layer designs, and applies this methodology to two very different cross-layer design proposals for WSNs. Through this review, we identify the need for a standardized framework for cross-layer design. Section IV lays out the basic requirements for a cross-layer framework, and briefly reviews some of the existing cross-layer framework proposals. Section V discusses the conclusions of this survey as well as lays out potential avenues of future work.

II. BACKGROUND

A. The OSI Model

In the late 1970s, as data communications became more widespread, designers recognized the need for a standardized development approach. The International Standards Organization kicked off the creation of such a standard in 1979, calling it the the *Open Systems Interconnection* (OSI) model [3]. The basic idea was to provide a common *framework* in which engineers could create interoperable communication protocols. By the late 1980s, the OSI Model had evolved into a widely adopted approach to communications system design.

The model is comprised of a set of "layers," each of which are responsible for specific functionalities of a communication system. The model also specifies the allowable interactions, or procedure calls, between every set of adjacent layers. Interaction between non-adjacent layers is never allowed. There are seven layers in total, some of which are not strictly required for specific designs.

Figure 1 illustrates the OSI seven layer protocol stack for two users and a data relay. We provide a brief review of the OSI layers, top to bottom, which will be useful for subsequent discussion:

1) Application: Handles application services, and as such is responsible for direct interaction with the user-defined applications.

2) Presentation: Converts between application data format and network data format, as well as handling data encryption.

3) Session: Opens, manages, and closes connections with other users (by interaction with their Session layer).

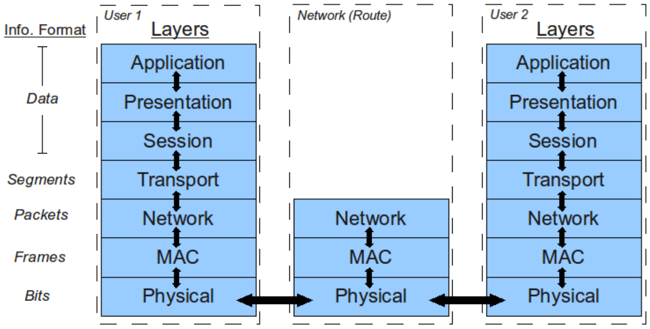

Fig. 1. The OSI Reference Model

4) Transport: Fragments data segments in network packets, and handles the error-free transmission and reception of these packets.

5) Network: Handles the conversion between the logical destination address, as specified by the application, and the physical destination address. Additionally, this layer determines routing decisions for the packets.

6) MAC/Data Link: Responsible for data flow over a network. Handles *channel access*, i.e. when to send information over the network, as well as converts and buffers packets into data frames. Frame-level error prevention, detection, and correction is handled at this layer.

7) Physical: Handles all aspects of the physical communication of a bitstream. Data en/decoding, de/modulation, and physical transmission/reception occurs at the physical layer.

The OSI standard has led to efficient protocol design because of the strict, clear boundaries between the layers. Each layer makes no assumptions about the adjacent layers, other than that they will be capable of pre-defined interactions as specified by the OSI standard. Thus, a single layer can be upgraded or replaced without affecting the design of the other layers.

Over the past twenty years, this approach has been shown to produce modular, robust, and enduring communication systems. In [2], the authors present historical examples of well-defined architectures which have led to long-term success for a number of different applications.

As wireless communications become the common mode of data transfer, the OSI model is beginning to show shortcomings. Researchers have demonstrated that joint optimization between multiple layers can lead to performance gains in wireless systems [1]. In order to make these optimizations, internal information and control must be shared between layers, which by definition violates the OSI standard. To understand the need for these modifications, we go back to first principles of communication channels.

In most wire-line systems, the channel models and physical architectures are well-understood and generally remain invariant. As such, the communication stack is optimally designed for the channel, and applications are designed to remain within the capability of the system. In wireless communications, the random time-varying nature of the channel inherently leads to varying performance of the communication system. This randomness can affect all of the OSI layers, and has been shown to cause degradation of system performance. While the OSI approach does have the advantages discussed above, it does not lend itself to highly optimized wireless system design.

A simple example of the wireless channel's effect on a communication system is the TCP protocol, which functions at the OSI Transport layer. Among other things, this protocol manages congestion in internet communications. When packets are dropped, the TCP protocol makes the assumption that the network is congested and abruptly lowers it's data rate. It then conservatively increases the rate over a significant period of time. This is a reasonable assumption and response for a wired Ethernet link, since bit error rates due to channel variation are extremely low.

In wireless systems, it is not uncommon to lose entire packets due to severe channel fading or interference. The TCP protocol views these dropped packets as signs of congestions regardless of the true cause, and backs off the packet-generation rate for a significant period of time. Meanwhile, the channel may have recovered after the loss of only a few packets. Thus, the TCP congestion response can lead to inefficient bandwidth utilization in wireless systems.

If there were a way for the TCP protocol to differentiate between dropped packets due to channel fading or congestion, this issue could be resolved. This solution was suggested and implemented. TCP packets now can include the Explicit Congestion Notice (ECN) bit, which notifies the TCP protocol when there is network congestion. Depending upon the state of this bit, the protocol reacts differently to a lost packet. This is a very simple example of what is generally termed *Cross-Layer Design* (CLD). In this case, the Transport layer (TCP) pays attention to the state of the Physical layer [4], and it clearly allows for improved protocol performance. More complex cross-layer optimizations can lead to significant performance improvements [1].

B. Cross-Layer Design Defined

In [1], a general definition of a cross-layer design is given as any violation or modification of the layered OSI reference architecture. The intent of CLD, simply stated, is to exploit information from multiple layers to jointly optimize performance of those layers. As depicted in Figure 2, four basic classes of cross-layer interactions are commonly utilized. System designers implement most of this functionality through either dynamic or static methods.

Dynamic cross-layer designs respond to changing network conditions. Designers create new, non-standard interfaces between OSI layers, merge functionality of multiple layers, and/or jointly calibrate layers. The new interfaces expose internal information or control parameters which were previously not externally accessible, but are required for cross-layer optimization.

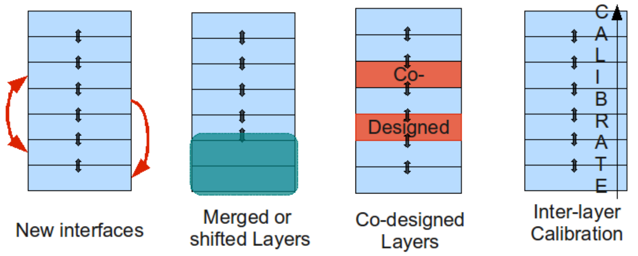

Fig. 2. Classes of Cross-Layer Design

For static cross-layer design, designers utilize known characteristics of the network and layers, and co-calibrate or co-design a set of layers off-line. By their very nature, all of these modifications destroy most of the benefit that the OSI framework provided. As such, designers must make careful decisions as to when, where, and how to implement cross-layer designs.

The authors of [2] summarize a number of potential issues with CLD:

1) Unintended Cross-Layer Interaction: The creation of new interactions between layers can lead to unforeseen dependencies which are not predicted by simulation. While designers make their best effort to vet system functionality through simulation and testing, real-world implementations are often subjected to unforeseen environments.

2) Stability: Stability of a given communication system is compromised by engaging in CLD. In the case of joint optimization, a given CLD may make changes at one layer based upon feedback from another layer. In effect, this creates a closed-loop feedback system, with all of associated typical design challenges. In communication systems, random variation occurs at the top (Application) and bottom (Physical) layers. Performance of any CLD must be carefully characterized against this system variation, which is often hard to capture, characterize, and simulate.

3) Long-term Sustainability: As a general rule, strict architectures with well-defined sub-system responsibilities and interactions lead to robust, modular designs. This is one of the greatest benefits of the OSI model. Each layer can be independently designed, changed or upgraded without any required action for the other layers in the system. The evolution of the internet is a great example of the success of this approach, as suggested in [2]. Clearly, cross-layer design severely impacts the modularity of any system, since layers now depend upon non-standard interfaces with other cross-layer optimized layers. Without careful consideration, a change made at any given layer could affect the functionality of any other layer. Additionally, it is not clear which, if any, CLD proposals could be combined to further improve performance.

III. CROSS-LAYER DESIGN BY EXAMPLE

We discuss some potential benefits of cross-layer design through two specific, distinct examples. WSNs are one of the strongest use-cases for CLD, because they are subject to the most stringent of application constraints. They are generally expected to have extremely long *network lifetimes*, and as a result must make efficient use of their limited, non-replenishable power sources.

Network lifetime is defined as the length of time for which a network maintains it's application-specified functionality. One sensor network might be required to last in the field for 10 years, while another may only need to work for 1 day. To illustrate the importance of energy-aware communication in wireless sensor networks, [5] notes that a WSN node designed by Rockwell, Inc. consumes approximately 2000 times more energy to transmit 1 bit than to execute a single CPU instruction.

We survey a few examples cross-layer design in literature for two purposes. First, it will clearly demonstrate the necessity for CLD in sensor networks. Second, as each proposal is reviewed, we will review it with a common methodology, such that we can compare the functionality of and understand the potential interplay between proposals.

A. Evaluation Criterion

Across the seven layers of the OSI stack, researchers have proposed many cross-layer optimizations. Cross-layer design proposals are generally specified in two forms of application requirements:

1) *Optimization objectives*, given a set of
2) *System constraints*.

An optimization objective might be network lifetime, as discussed above. In wireless ad-hoc networks, a common constraint is *flow conservation*, where each node must support an output data rate equal to the sum of self-generated outgoing packets and forwarding packets from other nodes. In general, the constraints are either *constructive* or *destructive*.

We define constructive constraints as those which provide relaxations such that the system can provide more optimization gain. In sensor networks, spatial correlation of data over an area can be exploited to provide a constructive constraint, as demonstrated in section III-B. Destructive constraints have the opposite characteristic, whereby they cause the system to have lesser optimization gain. The aforementioned flow conservation is an example of a destructive constraint. If we relaxed this and did not require flow conservation, the perceived, optimized network lifetime would be ”longer.” However, the data sink would never receive all of the information in the network, and would constantly be operating in a state of ever-increasing backlog! Clearly, this would be an undesirable system.

In evaluating each CLD proposal, we consider the following criterion:

1) Define the involved layers
2) Check system model and assumptions
3) State the Optimization Objectives
4) State the system constraints, constructive and destructive
5) Explain the nature of the optimization
6) Define new requirements for each involved layer

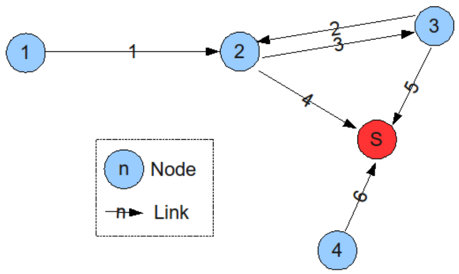

Fig. 3. Example Network Topology

B. Proposal: Optimization of Transmission Schemes

In [6], the authors propose a method for optimized data transmission in energy-constrained sensor networks. They begin by developing a model and set of assumptions for the system. Sets of potential optimization objectives and system constraints are defined. First, each of the three layers is individually optimized for three different objectives while the other two are held static. Next, all three layers are jointly optimized for a Time Division Multiple Access (TDMA) network, for cases where interference is and is not allowed. Computational results are compared, which demonstrate the gains achieved by cross-layer design. We focus on the cross-layer design proposals in this paper.

1) Involved Layers: Physical, MAC, and Network

2) Model & Assumptions: The authors assume a time-invariant network topology, such that the node links and link gain remain constant over the lifetime of the network. The channel noise is modeled as Additive White Gaussian Noise (AWGN). We describe the mathematical model for the network here, since it is a commonly utilized model in sensor network research. The network topology is described by a directed graph $\mathcal{G} = (V, L)$, where V is the set of nodes and L is the set of links. The graph is characterized by an incidence matrix $A \in \mathbb{R}^{VxL}$ and link gain matrix $G \in \mathbb{R}^{LxL}$. In figure 3, we provide an example network topology. Pictorially, arrows are links, where the arrow-head points to the receiving node. Entries in A are defined as

$$A_{v,l} = \begin{cases} 1 & v \text{ is the transmitter of link } l \\ -1 & v \text{ is the receiver of link } l \\ 0 & \text{otherwise} \end{cases}$$

The link gain matrix G describes the relationship between the set of links L. Entries on the diagonal, G_{ll} captures the power gain along a single link, which we desire to be large for a strong link. The non-diagonal $G_{lk}, (k \neq l)$ entries capture interference between links.

Additionally, as depicted in Figure 3, the authors assume a *single commodity flow*, such that all data generated by the network is destined to a single data sink. All nodes are assumed to have the ability to synchronize with each other, such that time is considered in *time slots* m_i of equal length. First, optimization within a TDMA network with orthogonal (non-interfering) links is considered. Then, a network with interference is considered.

A single, constant *bit error rate* (BER), $P_{b,req}$ is considered for the entire network, such that the data rate over a given link is bounded by a function of the BER and the *signal-to-interference-plus-noise* (SINR) ratio, γ_l. The bounding function here is defined as a standard model for MQAM communications. The power consumption for an active link in a time slot m is modeled as the sum of the node transmitter circuit and amplifier gain consumption. It does not include receiver circuit power consumption, which may not generally be a good assumption.

As shown in [7], consideration of receiver decoder complexity better optimizes network lifetime. Without considering receiver power consumption, the transmit power is reduced towards the capacity limit for a given link and rate. As transmit power and therefore received SNR decrease, the receiver decoder must iterate more in order to correctly decode the content of the received signal. With newer coding schemes like turbo codes, this decoder energy cost can be significant. Thus, depending upon the modulation and coding scheme utilized, consideration of receiver power consumption can further increase network lifetime.

3) Optimization Objectives: The authors study three different optimization objectives are studied:

- Total Power Consumption: For a case where there is ubiquitous deployment of sensor nodes, not every node is necessarily required in order for the network to function. Thus, through minimizing the total power consumption of the network, individual average node energy consumption is minimized fairly well in relation. This does not guarantee that every node survive for the entirety of the network lifetime. In fact, a weakness is identified in this approach. Since nodes closer to the sink tend to support higher data rates, they will die out more quickly than nodes further away from the sink. This leads to network fragmentation and loss of flow conservation. They suggest a weighted sum of ”node importance” can be applied to increase network lifetime in this optimization scheme.
- Minimum Node Lifetime: When all nodes in a sensor network are critical to its functionality, the objective is to maximize the minimum node lifetime. Additionally, this metric can be utilized in the case where there are many redundant network nodes. The optimization can be computed once at network start-up, and then be recomputed after enough nodes have died such that the network topology is significantly changed.
- Concave Functions of Node Lifetimes: In this optimization scheme, the network lifetime is considered as a concave function. Each node is considered to have a lifetime equal to the length of time for which its flow is supported. In other words, a node is considered alive so long as every piece of data it generates can reach the data sink, without causing backlog in the network. Additionally, this objective can exploit spatial correlation of data, by splitting the network into subregions with multiple nodes. So long as one node in that block can sense and report data, the network is considered functional. Thus, they

share responsibility for sensing data in their region and pool their energy resources, which increases total network lifetime.

4) System Constraints: The authors consider three system constraints, where some apply only in the interfering-link case.

- Data Rate: The maximum data rate, r_l, supported over a link, l, is bounded by a function $f_{rate,max}(P_{b,req}, \gamma_l)$. This is also a function of time (m). In this system, the rate constraint is chosen to represent M-ary Quadrature Amplitude Modulation (MQAM) communications. This constraint is required to ensure error-free reception of data at the receiver on link l. This constraint is directly considered only for the case where interfering links are present. In the non-interfering case, a single node has a simple relation between rate, flow, and number of allocated time slots. The energy consumption constraint discussed below captures the flow and time slot parameters. Rate can be determined from these two parameters, thus the constraint need not be considered when there is no interference present. This is a *destructive constraint*.
- Energy Consumption: This constraint captures the lifetime for a single node, or the average lifetime of a node depending upon the optimization objective. It takes slightly different forms in each of these cases. Essentially, the average power consumption for a node must remain below the total initial energy, E_i of that node, divided by the lifetime of the network. In other words, the average power consumption is upper bounded by the maximum plausible average power consumption, given the network lifetime. This is a *destructive constraint*.
- Flow Conservation: Flow conservation, as discussed above, requires that each node be able to support an outgoing data rate equal to the sum of incoming rate of forwarding packets and it's self-generated packets. This is a *destructive constraint*.

5) Optimization Methodology: Cross-layer optimization is considered for the three objectives listed above, in a slotted non-interfering TDMA network and/or in a slotted network with interfering links. First, the authors consider the TDMA case for lifetime maximization, subject to flow conservation, energy conservation, and other housekeeping constraints. The solution yields an optimal variable-length TDMA scheme. Next, the total network power minimization scheme is considered, where solutions are found for a special case of *identical equidistant nodes*. In general, this is a fair assumption for finding an approximate optimization routine in homogeneous sensor networks.

Finally, the case for a slotted network with interfering links is studied. This problem is formulated and found to be non-convex, and is therefore difficult to solve. By making a change of variables to alter the rate constraint, the modified f_{rate} function becomes convex. As such, the problem becomes a convex optimization and can be approximately computed. An algorithmic approach to find an approximate optimal solution is proposed, such that the central controller in the network can iterate over time slots to find an optimal rate, power, and scheduling solution.

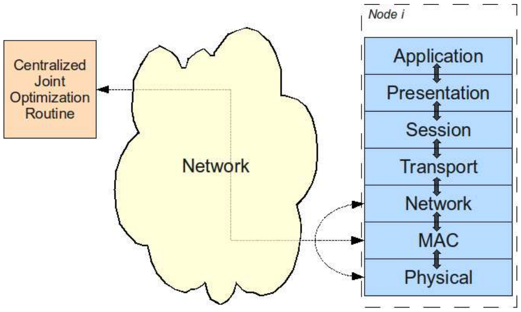

Fig. 4. Effect of Transmission Scheme Proposal on OSI

This algorithm is tested through an example case and is shown to improve network lifetime by 12% in comparison to three other routines. These gains are attributed to smarter routing at the network layer via energy-efficient multi-hop and load-balanced routes. Additionally, re-use of time slots contributes to network lifetime gains, since spatially distant nodes can share the same time slot, thus transmitting for more time slots at a lower average rate and power.

Note that this scheme does require centralized control, though it can terminate as soon as a nearly optimal solution is found. This iteration and dissemination of control through the network costs energy and thus impacts lifetime. Also, in the case where the source rates are non-stationary, i.e. the average source rate from a node drifts over time, the algorithm would have to run again to re-calibrate for new topology characteristics.

In Figure 4, we illustrate the interaction between the centralized CLD algorithm and an individual node communication stack.

6) Cross-Layer Method & Interfaces: The central algorithm iteratively computes rate, power, routing, and scheduling for each node. At the physical layer, rate and power control must be externally controllable. At the MAC layer, the scheduling must be externally controllable, although this action is always performed in TDMA networks. At the routing layer, the routing tables must be externally programmable. We note that this cross-layer design takes the form of dynamic vertical calibration between the Physical, MAC, and Network layers, where the central controller is responsible for updating the calibration of every node.

C. Proposal: Adaptive Quality of Service

We now evaluate another CLD proposal with very different goals and structure. In [8], the authors propose a CLD scheme for data reporting, which aims to balance application requirements and resources in order to provide an adaptive quality of service (QoS). Within the context of a QoS-aware sensor network communication system, the paper focuses on the design of three particular functions: *Data Reporting Tree Construction*, *Sensor Selection*, and *Data Report Scheduling*.

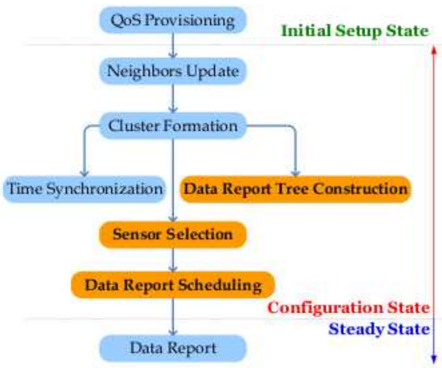

Fig. 5. QoS-Aware Cross-Layer System [8]

Figure 5 shows the overall system functionality, where the blocks of interest are colored orange.

1) Involved Layers: MAC and Network

2) Model & Assumptions: The sensor network model is defined as an undirected graph $\mathcal{G} = (V, E)$, where V is the set of sensor nodes and E is the set of undirected wireless links between nodes. Each node determines it's local neighboring links periodically. The WSN is defined to be a single-hop, cluster-based topology, where the set of densely deployed nodes are grouped in *clusters*. The clustering protocol is not covered in this paper.

Each cluster of nodes is composed of a cluster head and a multitude of members, where member nodes must be one hop from the head. The cluster head is responsible for a number of activities both within the cluster and within the overall network. It manages intra-cluster time synchronization, TDMA control, and data fusion, as well as inter-cluster network traffic control. The overall network operates as a TDMA-based system with spatial-reuse of time slots in different clusters. Each cluster head is responsible for mitigating neighboring cluster time-slot interference.

Within the entire network, there exists at least one data sink. Cluster heads are aware of all data sinks, as well as routing options to each sink. The cluster-heads organize themselves itself into a multi-headed data reporting tree, where the top of each tree is a data sink s. In order to pass information between cluster-heads, they identify and use neighboring members from different clusters to relay information. In other words, neighboring cluster-heads utilize a three-hop routing path to relay information across neighboring clusters.

D. Adaptive Quality of Service

1) Optimization Objectives: The authors define the single optimization objective as energy efficiency, such that minimum node lifetime is maximized. Cluster-heads are expected to make the most energy-efficient decision at the MAC and Network layers, given particular system constraints.

2) System Constraints: The proposal lays out two system constraints - one for the Network layer and one for the MAC layer.

- End-to-end delay: The user or application determines a delay constraint τ on the amount of time it takes for sensed data to reach a data sink, such that
$$\tau \leq \mathcal{D}_{rep}.$$
$\mathcal{D}_{rep}$ is defined as the total data reporting time, where it is defined as a function
$$\mathcal{D}_{rep} = f(\mathcal{D}_{prop}, \mathcal{H}_{int}, \mathcal{D}_{proc}, k).$$
System constant $\mathcal{D}_{prop}$ is the one-hop propagation delay. System constant $\mathcal{D}_{proc}$ is the amount of processing time for data fusion per cluster head. $\mathcal{H}_{int}$ is the number of intermediary cluster-heads in the multi-hop routing between the cluster-head data source and the data sink. Finally, k is the number of nodes within the data source cluster. With these cluster- and system- parameters, an estimate of the total time $\mathcal{D}_{rep}$ to report information from any cluster-source to sink can be calculated. This is a *destructive constraint*.
- Spatial Correlation: When collecting data within a given cluster, spatial data correlation between nodes is considered. The vector of measurements from any two nodes can be used to calculate a spatial correlation coefficient, ρ_{ij}. Based upon an application-defined limit, this correlation can be exploited to limit redundant packets. We will show through the optimization methodology that this is a *constructive constraint*.

3) Optimization Methodology: In the network start-up phase, the application disseminates the delay constraint τ throughout the network. In order to minimize the delay or meet the delay constraint, the fewest-hop count is chosen for tree construction. Within the bounds of the delay constraint, a load-balanced tree is constructed and maintained in order to drive longer lifetimes for all nodes involved in inter-cluster communications.

Within each cluster, the cluster-head maintains a table of correlation coefficients between every node. If the correlation between a set of nodes is significant, these nodes are considered as a single block B_i. In each data-collection phase, one node from each block is chosen to report data. Over time, the cluster-head fairly utilizes both nodes in order to maximize average lifetime of each node. On a round by round basis, the cluster-head can choose a node based on time since a report from that node, or based upon the residual energy in that node. Essentially, the power resources from all nodes within a block are pooled to increase the lifetime of that block.

Additionally, by decreasing the number of nodes which communicate per collection round, the length of each TDMA segment and the overall amount of data is reduced. Clearly this will lead to better overall performance.

4) Cross-Layer Method & Interfaces: For this proposal, two different cross-layer methods are utilized. For the routing scheme, new interfaces into the Network layer are required, such that an optimal route to the data sink can be chosen. For the MAC scheme, the application layer is designed to

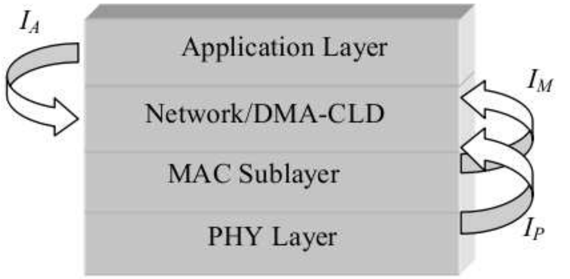

Fig. 6. DMA-CLD Framework [10]

specifically interact with a TDMA MAC, such that slots for a single node-per-block can be chosen. In general, this proposal is highly dependent on the rest of the communication stack, as discussed above.

E. Proposal Review

Here, we examined two very different CLD proposals. There are countless other examples of cross-layer design in literature.

In [5], the relationship between adaptive modulation, coding, and packet-delay and -rate constraints is studied in Rayleigh fading. The method exploits packet queuing in every node to avoid poor channel conditions, thus conserving energy. The cutoff SNR for transmission is modified based upon the packet-delay and -rate constraints. Another proposal in [9] employs utility maximization to optimize network performances. The upper protocol layers, which determine the utility function, select adaptive modulation and coding parameters for the physical layer. In [7], network lifetime is again maximized, this time utilizing the tradeoff between transmitter power and receiver power, based on coding complexity.

In each of these proposals different, overlapping sets of layers, assumptions, optimization goals, and system constraints are proposed. It is not clear how or which of these methods could be combined. There is no common methodology observed in developing or evaluating these ideas. It is clear that more organization would be useful in promoting efficient and universal CLD development.

IV. CROSS LAYER FRAMEWORKS

For practical communication systems design, the impact of the OSI Reference Model was unprecedented. For the last 20 years, it has provided a clear, universal framework for innovation and development. As wireless systems become the dominant mode of communication, a paradigm shift in design methodology is taking place. In communication systems design, it is no longer practical to consider purely separable sub-systems as the OSI Reference Model demands.

Even from a brief survey of existing cross-layer design proposals, it is clear that future CLD designs would benefit from a well-defined, widely-adopted framework. Development of a *Cross-Layer Framework* (CLF) along these lines has been proposed and explored in some depth, but no concrete, widely-supported effort has been undertaken [1]. None the less, as we move further from the OSI framework, we must begin to take steps towards a standard, cross-layer-capable framework.

In [1], three different classes of CLF proposals are identified. The first involves new interfaces for every layer of the OSI model, and removes the requirement that only adjacent layers communicate. The second class also utilizes an OSI-like stack, except new interfaces are only created between every standard layer and a single new shared layer, which would capture all of the cross-layer functionality. The third set of proposals involve entirely new abstractions for communication systems design. Each different class has potential benefits and downfalls. For context, we examine one specific CLF proposal from each class, and identify strengths and weaknesses of each.

A. CLF: Direct Inter-layer Communication

In [10], the Dynamic-Multi-Attribute Cross-Layer Design (DMA-CLD) framework was proposed. As depicted in Figure 6, this architecture utilizes an OSI-like stack, with new interfaces created between adjacent and non-adjacent layers. The framework is narrow in scope, since it only provides a cross-layer extension for optimized routing. New interfaces are created from the Application, MAC, and Physical layers in order to provide information to the DMA-CLD block, which resides in the Network layer. In fact, most of the functionality of DMA-CLD is implemented in the Network layer. The author states that DMA-CLD could be extended to interact with more layers and cover more optimization functionality, but does not actually extend the framework to this point.

This proposal, while novel in its time, is too limited to provide a broad, capable CLF. In general, direct inter-layer communication is not a good approach for CLD. The interaction between adjacent layers are well understood through the OSI model, and cross-layer interaction should take place outside of these. There are significant computational blocks associated with some cross-layer designs, and these should be implemented in their own sub-system.

However, this proposal does touch upon an important point. Both optimization objections and system constraints can be specified in terms of distributed node requirements or centralized network-level requirements. This is clearly evidenced through our earlier survey of CLD proposals, where a centralized algorithm was used for optimization, and the solution was disseminated throughout the network. As such, a capable CLF should be able to incorporate both levels of application characteristics.

B. CLF: Shared Super Layer

In [11], the Optimization Agent (OA) Framework is proposed. This architecture utilizes an OSI-like stack with a new shared super-layer, as shown in Figure 7. Each layer in the stack would provide new interfaces, such that cross-layer optimization would be offloaded to the OA super-layer.

This methodology is flexible, and should provide the ability to capture most CLD methods. If each stack layer has the correct interfaces to the OA, a CLD proposal which utilizes

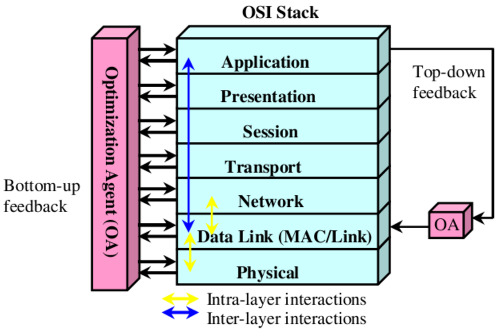

Fig. 7. The Optimization-Agent Framework [11]

those layers could function inside the OA. Of the classes of cross-layer designs specified in Section II-B, the OA can implement or mimic all of them. While layers can't be merged, all of the cross-layer required information and control could be shared between them via the OA.

In terms of system architecture, giving the OA unfettered access to every layer in dangerous, thus the new interfaces must be carefully defined and created.

C. CLF: New Abstractions

In [12], the TinyCubus framework is described. It was developed specifically with sensor networks in mind; the initial versions were built after evaluating a number of different WSN applications. It collapses the functionality and communication stack for a wireless sensor network into a three interdependent frameworks:

1) Data Management Framework: This is the heart of TinyCubus. Based on application-level requirements, system characteristics, and optimization objectives, it selects appropriate operating modes for each sensor node.

2) Cross-Layer Framework: The cross-layer framework incorporates sets of communication layers in order to create a functional communication system. The cross-layer interactions are provided through a state repository, which stores all of the necessary cross-layer parameters. Additionally, the data management framework accesses the cross-layer framework through this state repository. Which blocks are available and how these blocks are chosen is not entirely clear.

3) Configuration Engine: The configuration engine is responsible for network configuration and node updating. An important piece of this engine is the topology manager, which handles self-organization protocols for the network. Additionally, as wireless sensor networks are often subject to changing topologies, in some cases nodes may be expected to take on new functionality which was not previously envisioned. This engine provides a means for in-field node programming.

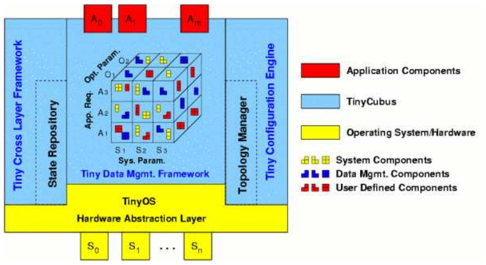

Fig. 8. TinyCubus Framework [12]

While it is geared specifically for wireless sensor networks, this CLF proposal has many attractive qualities. It incorporates the communication system, node functionality, and node management into a single framework for network operation and management. This is a logical choice, since sensor network communication and functionality are closely intertwined. Through the cross-layer state repository, the application and communication blocks can work together to provide system optimization. It is not exactly clear how a variety of CLD proposals would be cleanly incorporated into this framework, nor exactly how the application requirements and optimization goals should be chosen.

D. Review of CLF Proposals

In this section, we reviewed examples of the three different types of CLF proposals seen in literature. The latter two show promise for a universal CLF development. Both provide enough cross-layer interface capability such that most cross-layer optimizations are realizable. All of the reviewed proposals lack enough specification and definition to be considered a "standard."

Our list of criterion for CLD evaluation was intended to exemplify the need for a standard approach to CLD. For the creation of a universal CLF standard, more work and collaboration in this area is required. We must compile an exhaustive list of application requirements, optimization objectives, and system parameters for WSNs and other modalities of wireless communication systems. Additionally, a good CLF framework will define more than just communication blocks and allowable interactions. It will provide a methodology for selecting requirements and objectives, and suggest specific optimizations based on them. Thus, it will be more extensive than the OSI Reference Model.

The OSI Model was not developed in a vacuum. The ISO spearheaded efforts to bring together communication system designers and researchers in order to produce a flexible and useful framework for protocol design. A similar sort of joint effort will be required in order to produce a capable CLF.

V. CONCLUSIONS

As discussed in Section II, wireless communications are quickly becoming the dominant mode of data transfer. The OSI approach to communication systems design is not optimal for wireless communications. This is due to the random nature of the wireless channel. Through utilizing cross-layer design, researchers are creating "smarter" communication systems, which make programmed tradeoffs between application requirements in order to meet specific optimization goals. They exploit these tradeoffs to make better and more efficient use of the wireless channel. In light of many recent results showing the positive influence of CLD optimization, it has become an attractive and necessary design strategy.

In Section III, we reviewed some specific CLD proposals against some standard criterion. The first method jointly optimized the performance of the bottom three layers of the stack, by modeling those layers and subjecting them to a network lifetime maximization problem. The second described methods which conserve energy in the Network and MAC layers, against delay- and spatial correlation constraints. At the Network level, data reporting tree creation is subject to an application-specified delay constraint. Within every cluster, MAC scheduling is performed with knowledge of data correlation between nodes.

In Section IV we state the need for a standardized framework for CLD. We review the classes of CLF proposals, and examine some specific examples. We review the strengths and weaknesses of each, and determine that two proposed approaches for a CLF are viable. For a universal CLF development, wireless communication developers and researchers must come together to create a standard.

REFERENCES

[1] V. Srivastava and M. Motani, “Cross-layer design: a survey and the road ahead,” *Communications Magazine, IEEE*, vol. 43, no. 12, pp. 112–119, 2005. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2005.1561928

[2] V. Kawadia and P. R. Kumar, “A cautionary perspective on cross layer design,” *IEEE WIRELESS COMMUNICATION MAGAZINE*, vol. 12, pp. 3–11, 2005.

[3] M. D. Pardue, “Fine-tuning the osi model: Layer functions and services,” in *Military Communications Conference - Crisis Communications: The Promise and Reality, 1987. MILCOM 1987. IEEE*, vol. 1, Oct. 1987, pp. 0199–0203.

[4] S. Shakkottai, T. Rappaport, and P. Karlsson, “Cross-layer design for wireless networks,” *Communications Magazine, IEEE*, vol. 41, no. 10, pp. 74–80, Oct 2003.

[5] X.-H. Lin, Y.-K. Kwok, and H. Wang, “Cross-layer design for energy efficient communication in wireless sensor networks,” *Wirel. Commun. Mob. Comput.*, vol. 9, no. 2, pp. 251–268, 2009.

[6] R. Madan, S. Cui, S. Lall, and A. J. Goldsmith, “Modeling and optimization of transmission schemes in energy-constrained wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1359–1372, 2007.

[7] S. Vasudevan, C. Zhang, D. Goeckel, and D. Towsley, “Optimal power allocation in wireless networks with transmitter-receiver power tradeoffs,” in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–11.

[8] H. J. Choe, P. Ghosh, and S. K. Das, “Cross-layer design for adaptive data reporting in wireless sensor networks,” *Pervasive Computing and Communications, IEEE International Conference on*, vol. 0, pp. 1–6, 2009.

[9] D. O’Neill, A. Goldsmith, and S. Boyd, “Wireless network utility maximization,” in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, Nov. 2008, pp. 1–8.

[10] A. Safwat, “A novel framework for cross-layer design in wireless ad hoc and sensor networks,” in *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, Nov.-3 Dec. 2004, pp. 130–135.

[11] W. Su and T. L. Lim, “Cross-layer design and optimization for wireless sensor networks,” in *SNPD-SAWN ’06: Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 278–284.

[12] P. Marron, A. Lachenmann, D. Minder, J. Hahner, R. Sauter, and K. Rothermel, “Tinycubus: a flexible and adaptive framework sensor networks,” in *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, Jan.-2 Feb. 2005, pp. 278–289.